CLAIMS

1.   A plurality of computers interconnected via a communications link and operating at least one application program simultaneously.

2.   The plurality of computers as claimed in claim 1 wherein each said computer in operating said at least one application program reads and writes only to local memory physically located in each said computer, the contents of the local memory utilized by each said computer is fundamentally similar but not, at each instant, identical, and every one of said computers has distribution update means to distribute to all other said computers the value of any memory location updated by said one computer.

3.   The plurality of computers as claimed in claim 2 wherein the local memory capacity allocated to the or each said application program is substantially identical and the total memory capacity available to the or each said application program is said allocated memory capacity.

4.   The plurality of computers as claimed in claim 2 or 3 wherein all said distribution update means communicate via said communications link at a data transfer rate which is substantially less than the local memory read rate.

5.   The plurality of computers as claimed in any one of claims 1-4 wherein at least some of said computers are manufactured by different manufacturers and/or have different operating systems.

6.   A plurality of computers interconnected substantially as herein described with reference to Fig. 5 or Fig. 8 or Figs. 12-14 of the drawings.

7.   A method of loading an application program onto each of a plurality of computers, the computers being interconnected via a communications link, the method comprising the step of modifying the application as it is being loaded.

8.   The method as claimed in claim 10 wherein the modification of the application is different for different computers.

9.   The method as claimed in claim 7 or 8 wherein said modifying step comprises:-

(i)   detecting instructions which share memory records

(ii)   listing all such shared memory records and providing a naming tag for each listed memory record

(iii)   detecting those instructions which write to, or manipulate the contexts of, any of said listed memory records, and

(iv)   generating an alert instruction corresponding to each said detected write or manipulate instruction, said alert instruction forwarding the re-written or manipulated contents and name tag of each said re-written or manipulated listed memory record.

10.   A method of loading an application program onto each of a plurality of computers, said method being substantially as herein described with reference to Fig. 9 of the drawings.

11.   A method of operating at least one application program simultaneously on a plurality of computers all interconnected via a communications link and each having at least a minimum predetermined local memory capacity, said method comprising the steps of:

(i)   initially providing each local memory in substantially identical condition,

(ii)   satisfying all memory reads and writes generated by said application program from said local memory, and

(iii)   communicating via said communications link all said memory writes at each said computer which take place locally to all the remainder of said plurality of computers whereby the contents of the local memory utilised by each said computer subject to an updating data transmission delay, remains substantially identical.

12.   The method as claimed in claim 11 including the further step of:

(iv)   communicating said local memory writes constituting an updating data

transmission at a data transfer rate which is substantially less than the local memory read rate.

13.   A method of operating at least one application program simultaneously on a plurality of computers all interconnected via a communications link, said method being substantially as herein described with reference to Fig. 5 or Fig. 8 or Figs. 12-14 of the drawings.

14.   A method of compiling or modifying an application program to run simultaneously on a plurality of computers interconnected via a communications link, said method comprising the steps of:

(i)   detecting instructions which share memory records

(ii)   listing all such shared memory records and providing a naming tag for each listed memory record

(iii)   detecting those instructions which write to, or manipulate the contexts of, any of said listed memory records, and

(iv)   generating an alert instruction following each said detected write or manipulate instruction, said alert instruction forwarding the re-written or manipulated contents and name tag of each said re-written or manipulated listed memory record.

15.   The method as claimed in claim 14 and carried out prior to loading the application program onto each said computer.

16.   The method as claimed in claim 14 and carried out during loading of the application program onto each said computer.

17.   The method as claimed in claim 14 and carried out by just-in-time compilation.

18.   The method as claimed in claim 14 and carried out by re-compilation after loading.

19.   A method of compiling or modifying an application program to run simultaneously on a plurality of computers interconnected via a communications link,

said method being substantially as herein described with reference to Figs. 5 and 7-14 of the drawings.

20.   In a multiple thread processing computer operation in which individual threads of a single application program are simultaneously being processed each on a corresponding one of a plurality of computers interconnected via a communications link, the improvement comprising communicating changes in the contents of local memory physically associated with the computer processing each thread to the local memory of each other said computer via said communications link.

21.   The improvement as claimed in claim 20 wherein changes to the memory associated with one said thread are communicated by the computer of said one thread to all other said computers.

22.   The improvement as claimed in claim 20 wherein changes to the memory associated with one said thread are transmitted to the computer associated with another said thread and are transmitted thereby to all said other computers.

23.   A method of transmitting data between threads of a single application program simultaneously operating on a corresponding plurality of computers interconnected via a communications link, said method being substantially as herein described with reference to Fig. 10 or 11 of the drawings.